

Sockets Assignment 6 Report

Instruction:

Sockets assignment 6: threadedServer.py is a multithreaded server that can handle multiple clients at once, keeping a cumulative count of per-thread responses and total responses. Unfortunately sometimes the total response count gets messed up. Diagnose the problem and fix it. Do not change, move, or delete any existing lines of code; you may only add new code. The only system calls you may add are those in the `_thread` or `Threading` modules. In your report, include evidence that your modifications solve the problem. (est 1 hour)

Explanation:

The needed locks commands are allocate, acquire, release. Allocate will create locks, acquire will starts the lock, and release will terminate the lock that was initiated on release. The counter of cumulative response is messed up, thus I put the locks where the value `cumulativeResponses` is being used. Before assigning `oldCumulativeResponses` I put the lock (acquire), and after new value for `cumulativeResponses` is assigned I released the lock (release). In order to test, I combied the two lines of code:

```
oldCmulativeResponses = cumulativeResponses  
cumulativeResponses = oldCmulativeResponses + 1
```

To

```
cumulativeResponses += 1
```

And put the lock around it. It worked same as having two lines of code.

Code:

Client: (unchanged)

```
1. #!/usr/bin/env python3  
2. from socket import *  
3. s = socket(AF_INET, SOCK_STREAM)  
4. s.connect(("localhost", 7069))  
5. for x in range(1,5):  
6.     result = s.recv(512)  
7.     print(result)  
8.
```

Server:

```
1. #!/usr/bin/env python3
```

```
2. from socket import *
3. import datetime, time, random
4. import _thread
5.
6. def clientHandler(clientSocket, address, threadNumber):
7.     global cumulativeResponses
8.     myResponses = 0
9.     print("I'm a new thread, number %d" % threadNumber)
10.    print(" handling communications with ", address)
11.    for x in range(1,5):
12.        l.acquire()
13.        oldCumulativeResponses = cumulativeResponses
14.        time.sleep(2)
15.        time1 = time.time()
16.        nowtime = datetime.datetime.now()
17.        toSendString = "hello from " + gethostname() + nowtime.strftime(" %A %I:%M")
18.        toSendBytes = toSendString.encode()
19.        clientSocket.send(toSendBytes)
20.        cumulativeResponses = oldCumulativeResponses + 1
21.        l.release()
22.        myResponses = myResponses + 1
23.        print("Thread %d has done %d sends; all threads %d" %
24.            (threadNumber, myResponses, cumulativeResponses))
25.    clientSocket.close()
26.    _thread.exit()
27.
28. ##### main #####
29. nThreads = 0
30. global cumulativeResponses
31. cumulativeResponses = 0
32. l = _thread.allocate_lock()
33. s = socket(AF_INET, SOCK_STREAM)
34. s.bind(("localhost", 7069))
35. s.listen(5)
36. while True:
37.     nThreads = nThreads + 1
```

```
38.     c,a = s.accept()  
39.     _thread.start_new_thread(clientHandler,(c,a, nThreads))  
40.
```

Output:

```
Term2 Shell Edit View Session Scripts Profiles Toolbelt Window Help

Python
Python
Python
Python

handling communications with ('127.0.0.1', 57654)
I'm a new thread, number 3
handling communications with ('127.0.0.1', 57655)
I'm a new thread, number 4
handling communications with ('127.0.0.1', 57656)
Thread 1 has done 1 sends; all threads 1
I'm a new thread, number 5
handling communications with ('127.0.0.1', 57659)
Thread 1 has done 2 sends; all threads 2
Thread 1 has done 3 sends; all threads 3
Thread 4 has done 1 sends; all threads 4
Thread 2 has done 1 sends; all threads 5
Thread 5 has done 1 sends; all threads 6
Thread 3 has done 1 sends; all threads 7
Thread 1 has done 4 sends; all threads 8
Thread 4 has done 2 sends; all threads 9
Thread 2 has done 2 sends; all threads 10
Thread 2 has done 3 sends; all threads 11
Thread 2 has done 4 sends; all threads 12
Thread 4 has done 2 sends; all threads 13
Thread 3 has done 2 sends; all threads 14
Thread 3 has done 3 sends; all threads 15
Thread 3 has done 4 sends; all threads 16
Thread 5 has done 2 sends; all threads 17
Thread 5 has done 3 sends; all threads 18
Thread 5 has done 4 sends; all threads 19
Thread 5 has done 5 sends; all threads 20

Kens-MacBook-Pro:sockets.assignment6 ken$ python3 readfiveClient.py
b'hello' from Kens-MacBook-Pro, local Monday 10:58'
b'hello' from Kens-MacBook-Pro, local Monday 10:58'
b'hello' from Kens-MacBook-Pro, local Monday 10:58'
b'hello' from Kens-MacBook-Pro, local Monday 10:58'
Kens-MacBook-Pro:sockets.assignment6 ken$

Kens-MacBook-Pro:sockets.assignment6 ken$ python3 readfiveClient.py
b'hello' from Kens-MacBook-Pro, local Monday 10:58'
b'hello' from Kens-MacBook-Pro, local Monday 10:59'
b'hello' from Kens-MacBook-Pro, local Monday 10:59'
b'hello' from Kens-MacBook-Pro, local Monday 10:59'
Kens-MacBook-Pro:sockets.assignment6 ken$

Kens-MacBook-Pro:sockets.assignment6 ken$ python3 readfiveClient.py
b'hello' from Kens-MacBook-Pro, local Monday 10:58'
b'hello' from Kens-MacBook-Pro, local Monday 10:59'
b'hello' from Kens-MacBook-Pro, local Monday 10:59'
b'hello' from Kens-MacBook-Pro, local Monday 10:59'
Kens-MacBook-Pro:sockets.assignment6 ken$

Kens-MacBook-Pro:sockets.assignment6 ken$ python3 readfiveClient.py
b'hello' from Kens-MacBook-Pro, local Monday 10:58'
b'hello' from Kens-MacBook-Pro, local Monday 10:59'
b'hello' from Kens-MacBook-Pro, local Monday 10:59'
b'hello' from Kens-MacBook-Pro, local Monday 10:59'
Kens-MacBook-Pro:sockets.assignment6 ken$
```